Contents

Contributors xv
Preface xvii

Part 1 Principles of Parallel Computation 1

Chapter 1 Vector Processors and Multiprocessors 3
Walter J. Karplus *University of California, Los Angeles*

1.1 The Quest for Speed 3
1.1.1 Simulation of Distributed Parameter Systems 3
1.1.2 Real-Time Simulation of Dynamic Systems 4
1.1.3 Signal Processing of Sampled Data 4
1.2 Supercomputers and Minisupercomputers 5
1.3 Vector Processors: Design Alternatives 7
1.4 Peripheral Array Processors: Design Alternatives 9
1.5 Minisupercomputers: Vector Processors 11
1.6 Multiprocessors: Design Alternatives 14
1.7 Minisupercomputers: Multiprocessors 17
1.8 Performance Measures 19
1.8.1 Partial Differential Equations 22
1.8.2 Ordinary Differential Equations 25
1.8.3 Artificial Intelligence 26
1.9 The Changing Simulation Environment 28
Bibliography 29

Chapter 2 Exploiting Parallelism in Multiprocessors and Multicomputers 31
Kai Hwang *University of Southern California*

2.1 Supercomputer Architectural Advances 31
2.1.1 Evolution of Modern Supercomputers 31
2.1.2 Advanced Architectural Choices 34
2.2 Concurrency in Multiprocessors and Multicomputers 41
2.2.1 Processor Scheduling and Activity Control 41
2.2.2 Interprocessor Communications Schemes 43
2.2.3 Reconfigurable Interconnection Networks 48

2.3 Parallel Memory and I/O Multiprocessing 51
2.3.1 Memory Hierarchy and Access Methods 51
2.3.2 I/O Multiprocessing and Front End 56
2.4 Process Migration and Load Balancing 58
2.4.1 Process Migration Requirements 58
2.4.2 Dynamic Load-Balancing Methods 60
2.5 Performance Issues and Answers 61
2.5.1 Critical Issues to Achieve High Performance 61
2.5.2 Toward a Science of Parallel Computation 63
Bibliography 65

Chapter 3 Design Requirements of Concurrent Lisp Machines 69
Robert H. Halstead, Jr. *MIT Laboratory for Computer Science*

3.1 Introduction 68
3.2 Concurrent Lisp Languages 70
3.2.1 Unmodified Sequential Lisps 71
3.2.2 Side-Effect-Free Lisps 72
3.2.3 Lisp with Explicit Concurrency 73
3.2.4 Sources of Parallelism 76
3.2.5 Summary 78
3.3 Challenges to the Concurrent Lisp Machine Architect 78
3.3.1 Challenges from Lisp 79
3.3.2 Challenges from Concurrency 80
3.3.3 Challenges from Concurrent Symbolic Algorithms 82
3.3.4 Summary 83
3.4 Important Concurrent Lisp Machine Design Decisions 84
3.4.1 Scale of the Machine 85
3.4.2 Overall Organization 85
3.4.3 Addressing, Caching, and Replication 85
3.4.4 Storage Allocation and Garbage Collection 87
3.4.5 Latency Avoidance and Latency Tolerance 89
3.4.6 Summary 91
3.5 An Example 92
3.5.1 Overall Organization 92
3.5.2 The Communication Subnet 94
3.5.3 Memory Organization 95
3.5.4 Processor Organization 98
3.5.5 Summary and Discussion 101
3.6 Conclusions 102
Acknowledgments 102
Bibliography 103

Chapter 4 Design Issues of Multiprocessors for Artificial Intelligence 107
Benjamin W. Wah *University of Illinois at Urbana*
Guo-jie Li *Chinese Academia Sinica*

4.1 Introduction 107
4.1.1 Characteristics of AI Computations 107
4.1.2 Heuristic Searches 109
4.1.3 Faster Technologies and Parallel Processing 110
4.1.4 Design Issues of Parallel AI Architectures 112

4.2 Representation Level 113
4.2.1 Domain-Knowledge Representations 114
4.2.2 Metaknowledge Representations 117
4.2.3 AI Languages and Programming 120
4.3 Control Level 125
4.3.1 Consistency Maintenance 126
4.3.2 Partitioning 127
4.3.3 Synchronization 132
4.3.4 Scheduling 137
4.4 Processor Level 143
4.4.1 Microlevel Architectures 143
4.4.2 Macrolevel Architectures 144
4.4.3 Functional-Programming-Oriented System-Level Architectures 145
4.4.4 Logic and Production-Oriented System-Level Architectures 148
4.4.5 Distributed Problem-Solving Systems 149
4.4.6 Hybrid Systems 149
4.4.7 Fifth-Generation Computer Projects 150
4.5 Design Decisions of AI-Oriented Computers 152
4.6 The Future 154
Bibliography 155

Part 2 Advanced Computer Architectures 167

Chapter 5 Compute-Intensive Processors and Multicomputers 169
John L. Gustafson *Sandia National Laboratories*

5.1 Introduction 169
5.2 Applications of Array Processors 169
5.2.1 Seismic Data Processing 169
5.2.2 Medical Imaging 170
5.3 Applications of Attached Scientific Processors 171
5.3.1 Structural Analysis 172
5.3.2 Analog Circuit Simulation 173
5.3.3 Computational Chemistry 174
5.3.4 Electromagnetic Modeling 175
5.3.5 Seismic Simulation 176
5.3.6 AI Applications 177
5.4 Design Philosophy 177
5.5 Architectural Techniques 180
5.5.1 Functional Units 181
5.5.2 Processor Control 182
5.5.3 Parallelism and Fortran 185
5.5.4 Main Memory 185
5.5.5 Registers 187
5.5.6 Vector/Scalar Balance 187
5.5.7 Precision 188
5.5.8 Architecture Summary 191
5.6 Appropriate Algorithms 191
5.6.1 Dense Linear Equations 192
5.6.2 An Extreme Example: The Matrix Algebra Accelerator 193
5.6.3 Fourier Transforms 195
5.6.4 Finite Difference Methods 195
5.7 Metrics of Compute-Intensive Performance 197

5.8 Addendum: Hypercube Multicomputers 198
5.9 Concluding Comments 200
Acknowledgments 201
Bibliography 201

Chapter 6 Hypercube Systems and Key Applications 203
Yin L. Shih and Jeff Fier *Symult Systems Corporation*

6.1 Introduction 203
6.2 Example Hypercubes 205
6.2.1 AMETEK System 14 205
6.2.2 FPS T Series 207
6.2.3 Intel iPSC 210
6.2.4 NCUBE 211
6.2.5 Comparisons 214
6.2.6 Performance Measurements 215
6.3 Applications on Hypercubes 218
6.3.1 Separated Flow Simulations Using the Vortex Method 220
6.3.2 Transonic Flow 225
6.3.3 Fast Fourier Transforms 228
6.3.4 Implementations of the Alternating Direction Implicit Method 230
6.3.5 Solving Matrix Equations: LINPACK 234
6.4 Second-Generation Systems 239
6.4.1 Hardware Evolution 239
6.4.2 Software Revolution 241
Bibliography 243

Chapter 7 Parallel Architectures for Implementing Artificial Intelligence Systems 245
Kai Hwang *University of Southern California*
Raymond Chowkwanyun *Symult Systems Corporation*
Joydeep Ghosh *University of Texas at Austin*

7.1 Introduction 245
7.1.1 Symbolic Processing in AI 245
7.1.2 AI-Oriented Computers 246
7.2 A Taxonomy of AI Machines 248
7.2.1 Language-Based AI Machines 250
7.2.2 Knowledge-Based AI Machines 252
7.2.3 Connectionist Systems 254
7.2.4 Intelligent Interface Machines 257
7.3 Language-Based Machines 257
7.3.1 Lisp Machines: Symbolics 3600 Series 257
7.3.2 Prolog Machines: PIM in Japan 260
7.3.3 Functional Programming Machines: ALICE 262
7.4 Knowledge-Based Machines 266
7.4.1 Rule-Based Production Machines: DADO2 266
7.4.2 Semantic Network Machine: Connection Machine 270
7.4.3 Object-Oriented Machines: FAIM-1 273
7.5 Connectionist Systems and Intelligent Interface Machines 276
7.5.1 Connectionist Machine: Hypernet 276
7.5.2 Vision Machine: BBN Butterfly 280
7.6 The Future of AI Machines 282
Bibliography 284

Chapter 8 Comparison of the Cray X-MP-4, Fujitsu VP-200, and Hitachi S-810/20 289
Jack J. Dongarra and Alan Hinds *Argonne National Laboratory*

8.1 Introduction 289
8.2 Architecture 289
8.2.1 Cray X-MP 290
8.2.2 Fujitsu VP-200 (Amdahl 1200) 293
8.2.3 Hitachi S-810/20 295
8.3 Comparison of Computers 296
8.3.1 IBM Compatibility of the Fujitsu and Hitachi Machines 297
8.3.2 Main Storage Characteristics 297
8.3.3 Memory Address Architecture 300
8.3.4 Memory Performance 302
8.3.5 Input/Output Performance 304
8.3.6 Vector Processing Performance 306
8.3.7 Scalar Processing Performance 309
8.4 Benchmark Environments 310
8.5 Benchmark Codes and Results 311
8.5.1 Codes 311
8.5.2 Results 315
8.6 Fortran Compilers and Tools 315
8.6.1 Fortran Compilers 317
8.6.2 Fortran Tools 320
8.7 Conclusions 321
Acknowledgments 323
Bibliography 323

Chapter 9 Multicomputer Load Balancing for Concurrent Lisp Execution 325
Raymond Chowkwanyun *Symult Systems Corporation*
Kai Hwang *University of Southern California*

9.1 Introduction 325
9.2 Load Balancing in Multicomputers 326
9.2.1 Message-Passing Multicomputers 327
9.2.2 Control-Level Parallelism 328
9.3 Operating System Directives for Process Migration 329
9.3.1 Process Control Blocks 330
9.3.2 Operating System Directives 332
9.4 Hybrid Load Balancing Method 333
9.4.1 Sender- vs. Receiver-Initiated Modes 333
9.4.2 The Hybrid Method 336
9.4.3 Macro Dataflow Execution Model 337
9.5 Multicomputer Architectural Support 338
9.5.1 A Single Processor Architecture 339
9.5.2 The Postman and Awake Modules 339
9.5.3 Load Information and Decider Modules 341
9.5.4 Hybrid Software Architecture 342
9.6 Parallelism in Executing Lisp Programs 345
9.6.1 AND-Parallelism 345
9.6.2 OR-Parallelism 346
9.6.3 List and Argument Parallelism 346

9.7 Parallelization of Lisp Programs 348
9.7.1 Parallelization of AND/OR-Parallelism 348
9.7.2 Parallelization of List and Argument Parallelism 349
9.7.3 Parallelization of Gabriel Benchmark Programs 350
9.7.4 Compile-Time Overhead 354
9.8 Concurrent Lisp Benchmarking and Analysis 355
9.8.1 Gabriel Benchmark Results 355
9.8.2 Performance Analysis 361
9.9 Conclusions 363
Acknowledgments 364
Bibliography 365

Part 3 Parallel Processing Software 367

Chapter 10 Parallel Programming Environment and Software Support 369
Kai Hwang *University of Southern California*
Doug DeGroot *Texas Instruments*

10.1 Parallel Processing for Performance 369
10.1.1 Advanced Parallel Processing 370
10.1.2 Performance Benchmarking 371
10.2 Parallel Languages and Intelligent Compilers 371
10.2.1 Concurrent Programming Languages 372
10.2.2 Intelligent Compiler and Directives 374
10.2.3 Trace Scheduling for Scalar Parallelism 378
10.3 Vectorization and Multitasking Software 383
10.3.1 Vectorization and Migration Techniques 383
10.3.2 Multiprocessing and Multitasking 386
10.4 Parallel Algorithms and Applications Software 390
10.4.1 Mapping Algorithms onto Parallel Architectures 390
10.4.2 Mathematical Software and Application Packages 394
10.5 Creating a Parallel Programming Environment 398
10.5.1 User-Friendly Programming Environment 399
10.5.2 Techniques of Concurrent Programming 401
10.6 Conclusions 404
Bibliography 405

Chapter 11 Automatic Vectorization, Data Dependence, and Optimizations for Parallel Computers 409
Michael Wolfe *Oregon Graduate Center*

11.1 Introduction 409
11.2 Data Dependence 411
11.2.1 Data Dependence Directions 413
11.3 Vectorization Methods 416
11.4 Loop Concurrentization 423
11.4.1 Multivector Computers 426
11.5 Loop Interchanging 426
11.6 Applications to Parallel Languages 428
11.7 Computing Data Dependence Relations 430
11.7.1 Data Dependence Decision Algorithm 433
11.7.2 Examples of Dependence Computation 436
11.8 Summary 438
Bibliography 438

Chapter 12 Applying AI Techniques to Program Optimization for Parallel Computers 441
Ko-Yang Wang *Purdue University*
Dennis Gannon *Indiana University*

12.1 Introduction 441
12.1.1 The Trend Toward Parallelism 441
12.1.2 Automatic Program Parallelism Optimization 442
12.2 Abstracting the Machine Features and Building the Knowledge Base 444
12.2.1 Parallelism and Program Parallelization 445
12.2.2 Program Realization and Restructuring 446
12.2.3 Problems in Program Parallelism Optimization 447
12.2.4 Program Representation 448
12.2.5 The Representation of Machine Structures 449
12.2.6 Program and Machine Feature Abstraction 455
12.2.7 The Parallelism Metric 456
12.3 Intelligent Program Transformations 458
12.3.1 System Organization 458
12.3.2 Heuristic Hierarchy 459
12.3.3 Program Transformation with Heuristics 462
12.3.4 Organization of Transformation Heuristics 464
12.3.5 Applying a Transformation Hierarchy to Program Transformation 468
12.4 Conclusions 476
Appendix 478
Bibliography 483

Chapter 13 Restricted AND-Parallelism and Side Effects in Logic Programming 487
Doug DeGroot *Texas Instruments*

13.1 Introduction 487
13.2 Parallel Execution of Prolog Programs 488
13.3 Overview of the RAP Model 492
13.3.1 The Typing Algorithm 492
13.3.2 The Independence Algorithm 494
13.3.3 The Execution Graph Expressions 495
13.4 Examples of Execution Graph Expressions 496
13.5 Backtracking in the RAP Model 498
13.6 Side Effects 500
13.6.1 Side-Effect Goals—Some Terminology 501
13.6.2 Sequencing Side-Effect Goals 501
13.7 Compiling Side-Effect Expressions 503
13.7.1 Synchronization Blocks 505
13.7.2 Distributing Synchronization Blocks 507
13.8 Examples of Side-Effect Graph Expressions 510
13.9 Soft Side-Effect Built-Ins 518
13.10 Suspending Expression Execution 520
13.11 Summary 521
Bibliography 521

Part 4 New Computing Technologies 523

Chapter 14 Dataflow Computations for Artificial Intelligence 525
Jack B. Dennis *MIT Laboratory for Computer Science*

14.1 Introduction 525
14.2 Dataflow Computation 527
14.3 Artificial Intelligence 532
14.4 A Sentence Recognizer 533
14.5 Program Structure for the Search Process 539
14.6 Concurrency in Heuristic Search: The Parallel Priority Queue 544
14.7 Implementing Global Memory on a Dataflow Multiprocessor 550
14.8 Overall Performance Analysis 552
14.9 Conclusion 556
Bibliography 558

Chapter 15 VLSI Array Processors for Signal/Image Processing 561
S. Y. Kung *Princeton University*

15.1 Introduction 561
15.2 Mapping Algorithms to Systolic/Wavefront Arrays 563
15.2.1 Stage 1: Local DG Design 566
15.2.2 Stage 2: Signal Flow Graph Design 568
15.2.3 Stage 3: Systolic/Wavefront Array Processor Design 571
15.3 System Design Issues of Systolic/Wavefront Arrays 573
15.3.1 Organization of Systolic/Wavefront Array Systems 575
15.3.2 Matching Algorithms to Arrays 575
15.4 Signal/Image Processing Applications 585
15.4.1 Edge Detection by Means of Two-Dimensional Convolution 585
15.4.2 Shortest Path Problem 587
15.4.3 Dynamic Time Warping for Speech Recognition 594
15.4.4 Simulated Annealing for Image Restoration 597
15.4.5 Systolic Design for Artificial Neural Network 602
15.5 Concluding Remarks 606
Bibliography 607

Chapter 16 Exploring Neural Network and Optical Computing Technologies 609
Scott T. Toborg *Hughes Aircraft Company*
Kai Hwang *University of Southern California*

16.1 Introduction 609
16.1.1 Limitations of Conventional Computers 610
16.1.2 The Neural Network Paradigm 610
16.1.3 The State of Optical Computing 612
16.2 Principles of Neural Network Computing 613
16.2.1 Structure and Design Considerations 614
16.2.2 State Equations and Stability 617
16.2.3 Methods of Learning and Computation 619
16.3 Neural Network Models and Applications 620
16.3.1 Hopfield Model and the Traveling Salesman Problem 621
16.3.2 The Neocognitron and Character Recognition 624
16.3.3 Adaptive Resonance and Pattern Classification 628

16.4 Optical Computing Devices and Architectures 636
16.4.1 Optical Computing Devices 636
16.4.2 Architecture of an Optical Computer 638
16.4.3 Optical Crossbar and Hologram Networks 639
16.5 Optical Arithmetic Algorithms and Processors 641
16.5.1 Optical Symbolic Arithmetic 642
16.5.2 Totally Parallel Addition in Optics 643
16.5.3 An Optical Arithmetic Processor 647
16.6 Electronic and Optical Implementations 647
16.6.1 Fully Implemented Neural Network Chips 648
16.6.2 Virtual Coprocessors and Simulators 650
16.6.3 Optical and Electrooptical Implementations 650
16.6.4 Implementation Tradeoffs 654
16.7 Critical R/D Issues and Future Trends 654
16.7.1 Critical R/D Issues 655
16.7.2 Future Trends 656
Bibliography 657

Index 661